



# Enhancements & Extensions of Formal Models for Risk Assessment in Software Projects

Michael R. Murrah  
[mmurrah@nps.navy.mil](mailto:mmurrah@nps.navy.mil)

Luqi  
[Luqi@cs.nps.navy.mil](mailto:Luqi@cs.nps.navy.mil)

Craig S. Johnson  
[csjohnson@dcmdw.dema.mil](mailto:csjohnson@dcmdw.dema.mil)

Naval Postgraduate School  
2, University Circle  
Monterey, CA 93943 USA

## Abstract

Over the past 40 years limited progress has been made to help practitioners estimate the risk and the required effort necessary to deliver software solutions. Recent developments improve this outlook, one in particular, the research conducted by Juan Carlos Nogueira [1]. Dr. Nogueira developed a formal model for risk assessment that can be used to estimate a software project's risk when examined against a desired development time-line. This model is based on easily obtainable software metrics. These metrics are quantifiable early in the software development process.

Dr. Nogueira developed his model based on data collected from a series of experiments conducted on the Vite'Project simulation [2]. This unique approach provides a starting point towards a proven formal model for risk assessment, one that can be applied early in the software development lifecycle. Approaching software risk estimation has never previously been successfully accomplished in this manner.

The proposed research will provide definitive evidence that software risk assessment can be conducted early in software development using quantifiable metrics and simple techniques. Enhancements will be made to Dr. Nogueira's model, based on calibrations against post-mortem projects. These enhancements will result from many threads of research; extension of input metrics, increased number of simulation runs, simulation scenarios based on actual projects, and the introduction of a "gearing factor". Ultimately, the research will yield an improved risk assessment model, one that has been validated against thousands of post-mortem projects, having applicability to any software development activity.

## 1. Introduction

The current state of the art techniques of risk assessment rely on checklists and human expertise. This constitutes a weak approach because different people could arrive at different conclusions from the same scenario. The difficulty of estimating the duration of projects applying evolutionary software processes adds intricacy to the risk assessment problem.

## 2. Dr. Nogueira's Risk Assessment Model

Dr. Nogueira's research introduces a formal method to assess the risk and the duration of software projects automatically, based on measurements that can be obtained early in the development process. The method has been designed according to the characteristics of evolutionary software processes, and utilizes quantifiable indicators such as efficiency, requirement volatility and complexity. The formal model, based on these three indicators estimates the duration and risk of evolutionary software processes. The approach introduces benefits in two fields:

- a) Automation of risk assessment.
- b) Early estimation methods for evolutionary software processes.

Dr. Nogueira developed four software risk estimation models that show great promise in determining a software projects' associated risk early in the software development life cycle. The models accomplish early estimation by utilizing a set of quantifiable metrics that can be collected from the beginning of project development. In actuality, the requirements volatility metric is an estimation during the first development cycle and during subsequent development cycles is quantifiable. After each iteration of software development, the required

input metrics can be applied to the model in order to reduce the error in the model's results.

The minimum required input metrics, to support risk assessment, required for Dr. Nogueira's estimation model are the following:

a. Efficiency (EF) – The efficiency of the organization can be measured observing the fit between people and their roles [1]. Dr. Nogueira's research indicates that the efficiency of an organization can be directly calculated by computing the ratio of direct time (working and correcting errors) divided by the idle time (time spent without work to do).

b. Requirements Volatility (RV) – Requirements volatility expresses how difficult the requirement elicitation process is. The requirements volatility is obtained by the following formula [1].

$$\text{Requirements Volatility} = \text{Birth Rate Percentage} + \text{Death Rate Percentage}$$

Birth Rate Percentage (BR%) = the percentage of new requirements incorporated in each cycle of the software evolution process as calculated by:

$$\text{BR\%} = (\text{New Requirements} / \text{Total Requirements}) * 100 \text{ percent}$$

Death Rate Percentage (DR%) = the percentage of requirements that are dropped by the customer in each cycle of the evolution process as calculated by:

$$\text{DR\%} = (\text{Deleted Requirements} / \text{Total Requirements}) * 100 \text{ percent}$$

c. Complexity (CX) – Complexity has a direct impact on quality because the likelihood that a component fails is directly related to its complexity [1]. The complexity metrics can be determined in two forms: large granular complexity and fine granular complexity. These two forms of complexity can be directly determined from software specifications written in the Prototype System Description Language (PSDL) [3].

Large Granular Complexity (LGC) expresses the relational complexity of the system as a function of the number of operators (O), data streams (D), and types (T)

$$\text{LGC} = \text{O} + \text{D} + \text{T}$$

Fine Granular Complexity (FGC) expresses the relational complexity of each operator in the

system and is a function of the fan-in and fan-out data streams related to the operator [1]. For the purposes of the completed research and our notion of future research, the FGC metric is too specialized; our efforts concentrate on just the representation of the LGC.

$$\text{FGC} = \text{fan-in} + \text{fan-out}$$

Software developers can utilize Dr. Nogueira's four models to assess either the development time required to develop a project or determine the associated probability of completing a software project given the project's duration.

### 3. Previous Validation Research

In this section of the paper we present the results of validation attempts when using Dr. Nogueira's estimation models. The first is a result of the research conducted by Dr. Nogueira in his initial research and supplies data from simulations and comparisons to one project. The second validation endeavor is the results of research conducted on two additional projects [5].

#### 3.1 Dr. Nogueira's Validation

In conducting his research, Dr. Nogueira derived some initial conclusions with the models. The simulations showed that the three risk factors observed during the causal analysis (efficiency, requirements volatility, and complexity) have compound effects over the three parameters of the Weibull distribution [1].

Dr. Nogueira illustrates the results of the models against 16 simulated projects. Each model derives an increasing degree of accuracy based on: metrics from the three risk factors, Weibull cumulative density function, and the derivation of the time.

Models 1-2. Model 1 can be used when the requirements volatility is small. Model 2 considers the three factors (EF, RV, and CX), but neglects the combined effect of EF and RV. Figure 1 illustrates the results of the models which were calculated using 95% of confidence ( $p=0.95$ ). Note the errors as vertical segments between the estimated and real values.

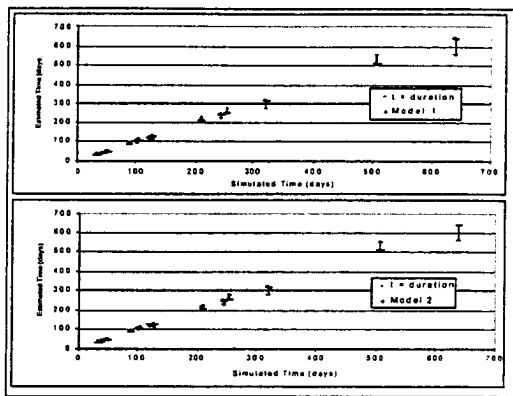


Figure 1. Scatter Plot of Models 1-2

Model 3. Model 3, illustrated in Figure 2, considers the three factors as well as the combined effects of EF and RV. The analysis of variance shows that the samples obtained from the simulations and the samples obtained from the estimates using Model 1, 2 or 3 cannot be statistically differentiated.

Another interesting result is that the errors remain in the range of  $\pm 15\%$  for all of the scenarios. This result is interesting if we compare it with the results of COCOMO ( $\pm 20\%$  in the best cases). Barry Boehm in reference to the validation of COCOMO said, "In terms of our criterion of being able to estimate within 20% of projects actuals, Basic COCOMO accomplishes this with only 25% of the time, Intermediate COCOMO 68% of the time, and Detailed COCOMO 70% of the time." [4].

Model 4. Model 4, Figure 2, can be used for any range of complexity and requirements volatility, and considers the three factors, their combined effects, and the following a priori assumptions:

- A project with 0 LGC will take 0 days
- $\alpha$ ,  $\beta$ , and  $\gamma > 0$
- If RV increases the  $p(x \leq t)$  decreases
- If CX increases then  $p(x \leq t)$  decreases
- If EF increases then  $p(x \leq t)$  increases

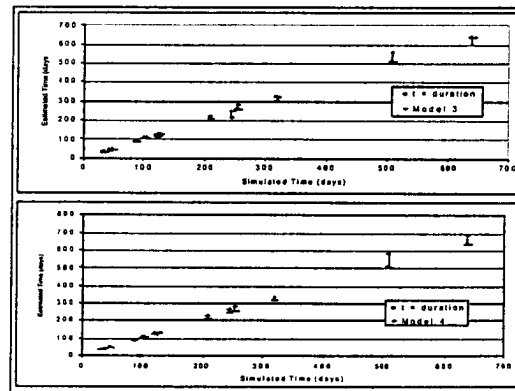


Figure 2. Scatter Plot of Models 3-4

The scatter plot in Figure 2 compares the simulated times versus the estimated times. Most of the errors are overestimations and the duration of the project has no effect over the percentage of error. Model 4 is conservative. The maximum overestimation error was less than 16% and the maximum underestimation was less than 4%.

Model 4 gives a good estimation for projects between 4,000 and 20,000 LGC (128 and 640 KLOC of Ada). The estimation seems to be too optimistic for projects smaller than 1000 LGC but it is quite good for larger projects. To verify the model Dr. Nogueira used a real project consisting of 1836 LGC developed in 1.5 years by the Uruguayan Navy<sup>1</sup>. Model 4 predicts 17 months instead of 18 months, the actual development time.

### 3.2 Additional Project Validation

Project A [5]. We used Nogueira's Model 4 to calculate the probability of completion curve for the projects. For consistency, we used working days, defined as 22 days per month, the same as used in the original Nogueira model.

The model predicted that the minimum time, in days, necessary to have a probability of completion of 100% is approximately 260 working days. When compared to the actual time it took, which was 336 working days, the model predicted completion sooner. The model predicted 76 working days less, or a 22.6% delta.

$$(1 - (260/336))(100) = 22.6.$$

At this point, with 22.6% variability, we decided to investigate and see what the original

<sup>1</sup> SIMTAS a simulator for war gaming with 75,240 lines of code

estimated completion date was from project records. The original estimation was 200 working days, with the project schedule slipping 136 working days for build 3. The developer missed the original completion estimation by 40.5%.

$$(1 - (200/336))(100) = 40.5.$$

The Nogueira model missed the developer's original estimate by 23.1%

$$(1 - (200/260))(100) = 23.1$$

Does this mean that the Nogueira model is too optimistic as are most developers' estimates, or is it a better fit? This data point leaves us with an inconclusive position as to the validation of the model against the first project. It appears that there is a difference when using real projects with real data versus simulated project data, and this reflects what the real world is – unpredictable.

**Project B [5].** We used Dr. Nogueira's Model 4 to calculate the probability of completion curve for Build 2 using; BR=2.59, DR=3.04, RV=5.63, O=2544, D=4010, T=1003. The model predicted Impossible.

Actual time for build 2 took from 4/24/00 until 7/10/00 or 68 working days at 22 working days a month. We believe this inconsistency is due primarily because the calculation for the LGC count is based on all six Computer Software Configuration Items (CSCI). Core functionality on three CSCIs; CSCI-A, CSCI-B, and CSCI-C had been previously developed and validated. However, the builds during this period, involved addition of functionality to the following CSCIs: CSCI-D, CSCI-E, and CSCI-F. That is, build 2 was modifying only a portion of the total software system code, but the LGC data gives a view of all six CSCIs combined.

The available data was not broken down into separate CSCIs, nor does it, post-mortem, identify the code that was being worked in a previous software release. We cannot fault the developer for not collecting metrics for research concepts that they are not aware of, nor do we believe that this type of data collection is a requirement of CMM level 3.

A finding of this research is the need to adjust the CX when applying the Nogueira model to evolved projects that are developing or enhancing only a portion of their CSCIs.

Additionally, this project did not utilize a lower case tool such as Rational Rose. We believe use of such a tool is essential when attempting to apply the Nogueira formal model, as it provides the capability to collect detailed information, over the software development lifecycle, that can later be extracted and used for input to the Nogueira model metrics.

#### 4. Issues with Dr. Nogueira's Risk Assessment Model

Applying Dr. Nogueira's risk assessment model, in its current form, presents a number of issues that must be resolved before substantial progress can be achieved validating the model's results. The first issue and most notable drawback when using Dr. Nogueira's risk assessment model is limited confidence that the model provides valid results. This is due to three factors: the limited amount of time that the model has been in existence, the model has not been exercised on a wide base of real world projects (completed or on-going), and the fact that the model was developed using simulation techniques. The first factor noted can only be dealt with in the passage of time. However, this research will exploit a unique opportunity to impact the latter two issues.

Although Dr. Nogueira's research shows promise in estimating the associated risk when developing software systems, the model has not been significantly exercised beyond theoretical simulation. Three 'real world' projects to date have been applied against the estimation model [1], [5]. It should be noted that all three of these projects were exercised post-mortem. Model validity has not been demonstrated in the context targeted by the model's original design, estimating risk early in a software project's life cycle.

A second issue that exist when using Dr. Nogueira's risk assessment model is the required input metrics. This issue is a double-edged sword. A major attraction to using Dr. Nogueira's model are these metrics. They are determined in a definitive, quantifiable manner and can be derived extremely early in the software development process [1], [6]. However, these metrics are quite unique. Currently, outside of the academic environment, it is not common practice to collect these unique metrics in the required form to utilize Dr. Nogueira's risk assessment model.

In order to establish confidence in the usefulness and accuracy of Dr. Nogueira's risk estimation model, the model must be exercised

against numerous projects. It would be ideal, and perhaps over time, to exercise the model according to its original design; early in the software development cycle. However, the next logical step is to continue to exercise the model in a post-mortem basis. Before this can be accomplished, two things need to happen: First correlations must be determined between Dr. Nogueira's required metrics and metrics that are frequently collected in historical project databases. By establishing metrics correlations, the model can be exercised against an additional project base helping address the second factor of problem one. And second, a method other than the use of PSDL to generate O, D and T metrics counts must be developed. Dr. Nogueira's model was based on using PSDL to automatically scan and generate counts for O, D, and T input to his model. It is unlikely that PSDL was used on any programs that we have post-mortem data on.

The final problem associated with Dr. Nogueira's risk assessment model is the configuration of the Vite'Project simulation. Dr. Nogueira developed the configuration of Vite'Project using Organizational Consultant expert system. Fictitious software engineering organizations were developed to represent the typical software development department. Based on the results of establishing fictitious CMM level 2 and level 3 organizations, the Vite'Project was calibrated. Calibrating the simulation in this manner, could yield different results than calibrating the simulation with actual information derived from real projects. If Dr. Nogueira's model can be verified by reprogramming the Vite'Project configuration this would provide additional assessment to the third factor of problem one.

## 5. Proposed Research

The proposed research will expand the efforts of the previous validation effort. Figure 3 outlines the research approach.

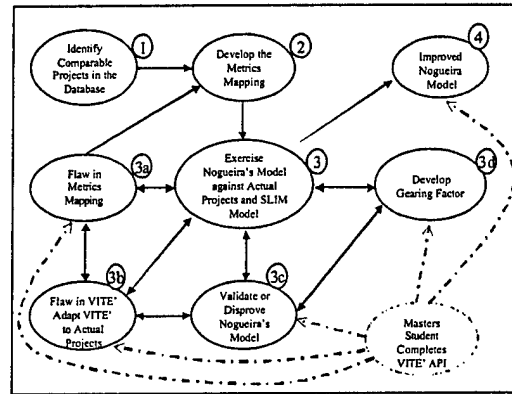


Figure 3. Phases of Research

Phase one: During phase one of the research, post-mortem projects will be identified whose characteristics are similar to the characteristics of the three projects previously exercised against Dr. Nogueira's risk assessment model. This affords the opportunity to begin with a baseline before proceeding to future phases.

Phase two: This is the most challenging phase of the research and we hypothesize that this phase will consume the majority of the available resources. In this phase, detailed analysis is conducted against the available metrics that have been collected on the projects established during phase one. Correlations are determined in the available data against the three metrics that are necessary when utilizing Dr. Nogueira's model. Upon completion of this phase, when a suitable "metric map" has been developed, research can continue to phase three. The intent of the metric map is to provide a common platform to exercise Dr. Nogueira's model using metrics that were not originally collected for this purpose.

Phase three: Once a suitable metric map has been established, research continues by exercising Dr. Nogueira's model against the set of post-mortem projects determined in phase one. This phase is essential to establish confidence in the results produced when using Dr. Nogueira's model. Additionally during this phase, another risk assessment method is introduced, Quantitative Software Management's® (QSM) SLIM, to help in the validation process. Essentially, there will be a comparison of three artifacts: the recorded project performance, the estimated project performance using Dr. Nogueira's model, and

the estimated project performance as determined by QSM's SLIM. An assumption during this phase will be the accuracy of QSM's SLIM. Of course, if the expected results are not achieved during this phase, additional research must be performed to determine the cause of the variance.

Phase three (a): One potential cause of the variance observed during phase three could be a flaw in the metric map determined during phase two. Continued research will be conducted to modify the mapping and eventually minimize the chance that the metric map is the source of the deviation.

Phase three (b): Another factor that can influence deviation between the actual project data, Dr. Nogueira's estimation model, and QSM's SLIM estimation model is the original configuration used to establish project scenarios in the Vite'Project. Organizational Consultant expert system was used to establish fictitious software engineering organizations. Research may indicate that reprogramming the Vite'Project with actual information from software development organizations could yield different results in the Vite'Project simulation. This was a fundamental factor in the development of Dr. Nogueira's research. A substantial change in the simulated results could require extensive rework of Dr. Nogueira's model.

Phase three (c): Finally, after exhausting Phases three (a & b), research may lead to examination of Dr. Nogueira's model with closer scrutiny. If deviation continues to present itself when conducting phase three, we may have essentially resort to "ground zero" to establish potential conflicts.

It should be noted that phases three (a, b, & c) should not be considered mutually exclusive. Research could indicate that partial modifications are required in all three sub-phases.

Phase three (d): Dr. Nogueira's risk assessment model is perfectly suited for any evolutionary software process because it follows the same philosophy [1]. Dr. Nogueira presents no hypothesis of the model's validity when the model is exercised outside of this domain. Once phase three is accomplished and confidence has been established against the set of projects determined during phase one, the model can be exercised against additional projects, from different industry sectors and different software development methodologies. This may require the development of what we are calling a

"gearing factor". In this research, the use of this term is intended to represent a value that is multiplied by the results determined in Dr. Nogueira's model, adjusting the results for the new domain. In some cases the model may provide suitable results without the use of a gearing factor, other domains and development methodologies may require this adjustment due to the unique nature of the software's development.

Phase four: Phase four of the proposed research is the culmination of all of the proposed research. This phase delivers the improved Nogueira model. A caveat to this phase and all of the sub-phases conducted during phase three is the introduction of the Vite'Project API. This automated tool will improve the statistical significance obtained when utilizing the Vite'Project simulation, greatly increasing the number of simulation runs provided by the simulation.

## 6. Validation

We propose to validate our research by conducting controlled experiments against post-mortem projects. QSM, founded in 1978 by Larry Putnam, has collected and maintained an extensive database of over 5,000 software projects [7]. Experiments can be conducted, utilizing the available software metrics from QSM's database, that correlate the required metrics in Dr. Nogueira's model. This will afford our research the means to evaluate actual projects against Dr. Nogueira's model.

Another source of validation is obtained by configuring Vite'Project with actual software project development information. As previously mentioned, Vite'Project scenario's were originally established by the creation of fictitious software development organizations. Different results could be derived from simulations configured according to actual projects.

Finally, we propose to increase the statistical significance of Dr. Nogueira's software risk assessment model. We can accomplish this by increasing the simulation runs of each scenario through automation via the Vite' API when available.

## 7. Conclusion

This research introduces a research plan to validate a formal risk assessment model for software projects based on probabilities and metrics automatically collectable early in the project. The approach enables a project manager to evaluate the probability of success of the

project very early in the life cycle. For more than twenty years the estimation standards (COCOMO 81, COCOMO II, Putnam) have been characterized by a common limitation: the requirements should be frozen in order to make estimations. This promising model removes this important limitation, facing the reality that requirements are inherently variable.

The problem of risk assessment for projects has been treated as unstructured. Research shows, and experiments will prove, a structured method to solve the problem based on metrics automatically collected from the project baselines. This contribution impacts the software engineering state of the art, as well as risk management in general. These metrics measure three risk factors identified in the research: complexity, requirements volatility, and efficiency. The subjectivity issue characteristic of previous research has been eliminated. Any decision-maker will arrive at the same estimates, independently of his or her expertise.

Finally, current research is based on simulations and a small set of real projects. It is desirable to collect and analyze metrics and completion times of a larger set of real software projects to confirm and refine the models. Our research will provide the missing elements from the models, validation, enhancements, and extensions.

## References

---

- [1] Nogueira J.C., *A Formal Model for Risk Assessment in Software Projects*. PhD Dissertation. Naval Postgraduate School. Monterey, California. 2000.
- [2] The Vite'Project Handbook. Vite©. 1999.
- [3] Berzins, V. and Luqi. *Software Engineering with Abstractions*. Addison-Wesley, 1990.
- [4] Boehm, B. *Software Engineering Economics*. Prentice Hall, 1981.
- [5] Johnson, C. S., Piirainen R. A. *Application of the Nogueira Risk Assessment Model to Real-Time Embedded Software Projects*. Masters Thesis. Naval Postgraduate School. Monterey, California. March 2001.
- [6] Nogueira, J.C., Luqi, Bhattacharya, S. *A Risk Assessment Model for Software Prototyping Projects*. Rapid System Prototyping, 2000. RSP 2000. Proceedings. 11th International Workshop on , 2000 Page(s): 28 -33
- [7] SLIM MasterPlan User's Guide. QSM® March 2001.